Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.057710

ARTICLE

Malicious Document Detection Based on GGE Visualization

Youhe Wang, Yi Sun*, Yujie Li and Chuanqi Zhou

Henan Province Key Laboratory of Information Security, Information Engineering University, Zhengzhou, 450000, China

*Corresponding Author: Yi Sun. Email: 11112072@bjtu.edu.cn

Received: 26 August 2024 Accepted: 30 October 2024 Published: 03 January 2025

ABSTRACT

With the development of anti-virus technology, malicious documents have gradually become the main pathway of Advanced Persistent Threat (APT) attacks, therefore, the development of effective malicious document classifiers has become particularly urgent. Currently, detection methods based on document structure and behavioral features encounter challenges in feature engineering, these methods not only have limited accuracy, but also consume large resources, and usually can only detect documents in specific formats, which lacks versatility and adaptability. To address such problems, this paper proposes a novel malicious document detection method-visualizing documents as GGE images (Grayscale, Grayscale matrix, Entropy). The GGE method visualizes the original byte sequence of the malicious document as a grayscale image, the information entropy sequence of the document as an entropy image, and at the same time, the grayscale level co-occurrence matrix and the texture and spatial information stored in it are converted into grayscale matrix image, and fuses the three types of images to get the GGE color image. The Convolutional Block Attention Module-EfficientNet-B0 (CBAM-EfficientNet-B0) model is then used for classification, combining transfer learning and applying the pre-trained model on the ImageNet dataset to the feature extraction process of GGE images. As shown in the experimental results, the GGE method has superior performance compared with other methods, which is suitable for detecting malicious documents in different formats, and achieves an accuracy of 99.44% and 97.39% on Portable Document Format (PDF) and office datasets, respectively, and consumes less time during the detection process, which can be effectively applied to the task of detecting malicious documents in real-time.

KEYWORDS

Malicious document; visualization; EfficientNet-B0; convolutional block attention module; GGE image

1 Introduction

With the rapid development of the Internet and artificial intelligence, cyberattacks are constantly evolving, and malicious documents, as significant carriers of APT attacks, pose a severe threat to individual users, enterprises, and even national security. In 2024, researchers [1] at Fortinet FortiGuard Labs discovered a new banking Trojan called "CHAVECLOAK", which spread via phishing emails with PDF (Portable Document Format) attachments, tricking victims into clicking on and downloading embedded malicious links to steal sensitive financial-related information. Perception Point, an Israeli network security company, released a report pointing out that against the phishing activity

of American enterprises-Operation PhantomBlu [2], attackers used Microsoft Office documents containing NetSupport RAT Trojan horse to spread them through e-mail, inducing users to click, and then remotely stealing sensitive data. The latest report from Check Point Research [3] showed that nearly 70% of file-based email attacks globally utilize malicious PDFs, an increase of 20% year-over-year. The above data show that cybercriminals see PDF files as an effective way to spread malware. Therefore, how to quickly and effectively identify malicious documents has become a hot research issue in network security.

Many researchers have tried to solve this problem in recent years, and many new malicious document detection methods have been proposed. The traditional malicious document detection techniques mainly rely on signature matching and heuristic analysis methods. By adopting the idea of pattern matching, we extract the feature signature of malicious code from malicious documents and store it in the virus database to determine whether the document is infected by malicious code. However, despite evolving malicious code analysis techniques, these traditional methods show significant limitations, hardly responding effectively to complex and changing virus threats. To overcome these challenges, researchers have turned to machine learning techniques to extract byte-sequence document content, structural features [4,5], and statistical features [6–8] through static analysis from documents. Although this method improves detection accuracy, it still requires much human involvement to perform feature engineering, a complicated and time-consuming process. Furthermore, these methods are usually limited to detecting specific types of malicious documents and need more applicability and flexibility. In dynamic analysis, researchers extract and analyze information about the behavior of documents at runtime by executing them in a controlled environment [9–11]. Although dynamic detection can provide in-depth insights into document behavior, it consumes several computational resources; secondly, it is time-consuming to execute in a sandbox environment and complex to apply to large-scale detection.

Several researchers have used the information entropy to measure randomness and uncertainty to recognize malicious documents in recent years [7,12,13]. The techniques used by malicious document writers, such as code obfuscation, shell deformation, and insertion of NOP (No Operation) instructions, will leave traces of entropy distribution. Accordingly, the researchers adopt various technical means to extract key features from entropy sequences. The research results show that the features hiding in the entropy of malicious document information can effectively characterize the malicious code embedded in the document and play an essential role in identifying malicious code. Therefore, this paper conducts more in-depth research on entropy sequences to extract significant features that can represent the properties of malicious documents.

In the background of the fast development of image recognition technology and deep learning, the research community has taken new steps in the field of malware detection. Many researchers no longer focus on these non-visual detection methods but focus on malware visual classification techniques [14–16]. By visualizing the binary data of malware as images and combining the idea of transfer learning, researchers use deep neural networks (e.g., VGG16, InceptionV3, and ResNet50, etc.) to deeply explore the texture features in images to achieve the classification of malware [17,18], which not only avoids the dependence of manual feature extraction but also provides a new perspective for malware identification and classification. However, it is worth noting that, despite the remarkable achievements in the visual classification of malware, there has yet to be a good application of visualization techniques for malicious document detection.

Inspired by the above work, we proposed a malicious document detection method based on GGE visualization in this paper. By combining the malicious document visualization technique with

deep learning algorithms, we avoid the complexity of manual feature extraction while reducing the high resource consumption and time cost. We also profoundly explore the document visualization features beneficial for model classification. Generally speaking, the contributions of this paper can be summarized as follows:

a) We proposed a novel malicious document detection method that visualized documents as GGE images. GGE three-channel image consists of a fusion of grayscale image, grayscale matrix image, and entropy image. Compared with previous methods, we extracted features directly from the original byte sequence of the document rather than relying on the structural data or metadata of the document, which enabled it to deal with multiple file formats, improving the versatility and robustness of the detection method. Through comparison experiments, we find that GGE images show more significant superiority than converting malicious documents to grayscale images or directly mapping them to Red Green Blue (RGB) images.
b) We introduced a grayscale-level co-occurrence matrix (GLCM) visualization technique. Unlike traditional methods that calculated statistical features (e.g., contrast, energy, uniformity, and correlation) directly from the GLCM, we adopted a method to convert texture and spatial information features from GLCM into a visualized image, which was generated by extracting the grayscale co-occurrence matrices of the grayscale image in four different directions. The experimental results demonstrated that this approach had significant results in classifying malicious documents and played a crucial role in improving classification accuracy.
c) We applied pre-trained models from the domain of transfer learning to the malicious document detection task. By combining three different image visualization features with the CBAM-EfficientNet-B0 model, we deeply explored the byte-level data features of malicious documents and entropy sequence features. We fused the channel and spatial attention information of the feature maps to achieve a more accurate classification result for malicious documents.

Section 2 introduces related work in malicious document detection and visualization. Section 3 describes our proposed method, including the feature extraction process, the construction of GGE images, and the design of the improved model. Section 4 shows the experimental process and the analysis of the results. Section 5 summarizes and outlooks the method of this paper.

2 Related Work

In this section, the related research of malicious detection classification is presented, mainly including malicious detection methods based on static features, dynamic features, and visualization features.

2.1 Malicious Detection Based on Static Features

Currently, static analysis is divided into two main categories. The first category focuses on analyzing the content and structural features of documents. Šrndić et al. [19] combined structural paths with metadata as new features for better detection, but there are a lot of problems that need to be improved in this method, such as the abstraction of the feature is not deep enough, and the extraction of metadata features is not deep enough, the model relies on the open-source Poppler PDF rendering library for document parsing, but the tool's performance in parsing obfuscated malicious documents is not satisfactory, which may affect the model's accurate identification of malicious documents. Cohen et al. [20] proposed SFEM (Structural Feature Extraction Methodology), which extracted structural features from office documents. Li et al. [21] focused on extracting structural features from documents and used a Support Vector Machine and Random Forest as classifiers, improving

detection accuracy by combining active learning strategies. Researchers in the literature [22,23] used tools to extract static features such as PDF document content and structural keywords. They adopted machine learning models such as Random Forest and AdaBoost for classification to identify malicious documents. Wen et al. [24] combined PDF vulnerability analysis examples to construct vulnerability detection rule libraries for static rule matching, using stack, function call, memory, and dynamic link libraries techniques to detect ROP (Return-oriented Programming) chains, which could better deal with the variability of malicious PDFs while obtaining better vulnerability detection capabilities. These methods improved detection accuracy, but feature engineering is complex and requires manual design and selection of features.

The second category focuses on detecting the statistical features of documents. Jiang et al. [25] extracted statistical features of entropy sequences using wavelet energy spectrum analysis and combined them with structural features of PDF documents to identify and detect malicious PDF documents. Jeong et al. [6] input PDF document byte sequences into convolutional neural network to predict whether there was malicious behavior in given sequences or not. Liu et al. [12] and Guo et al. [7] divided the original byte sequence of files into multiple blocks and calculated the entropy value of each block to the information entropy sequence, respectively, and then extracted the structural features of the information entropy sequence based on Haar wavelet transform and bag-of-words model to realize the classification of malicious documents and malicious families. Zhou et al. [8] proposed a novel method that identified malicious documents by converting the binary differences between malicious and standard documents into power spectrum differences of the file entropy sequence. This method utilized power spectrum analysis to quantify and compare the entropy sequences of documents, revealing potentially hidden malicious features in the documents.

2.2 Malicious Detection Based on Dynamic Features

Malicious document detection methods based on dynamic analysis rely on executing the content of a document to observe its behavior, which can reveal potentially malicious behavior when the document is running. Carmony et al. [9] investigated the memory access patterns and execution paths of PDF documents when opened in Adobe Reader. They identified three key program memory locations associated with the extraction of JavaScript code, the termination of PDF document processing, and the handling of errors, thus capturing the JavaScript information contained therein. However, this approach has limitations in that it is restricted to extracting JavaScript code that Adobe Reader spontaneously executes. Xu et al. [10] developed the PlatPal system, which tracked and observed PDF behavioral patterns by opening PDF documents on different operating systems using the same PDF reader. This approach allows researchers to analyze the behavior of PDF documents from multiple perspectives, which improves the comprehensiveness and accuracy of detection but has excessive overhead and low detection efficiency. Jiang et al. [11] proposed a manual-free feature dictionary detection model called NFDD (No manual Feature Dictionary Detection), which combined word embedding-based neural networks with dynamic analysis techniques to capture behavioral information of unknown samples and enhance the detection of new and unknown malware. The methods based on dynamic analysis provide insights into document behavior and are particularly suitable for detecting documents that use sophisticated tricks to hide malicious intent. However, these methods consume computational resources and are hard to apply to large-scale detection.

2.3 Malicious Detection Based on Visualization Features

Visualization technology is widely used in computer security, especially in malware detection. The technology transforms the binary content of malware into images in a specific way, providing security

analysts with an intuitive and efficient means of detection. Zhao et al. [16] proposed MalDeep, a deep learning malware classification framework based on texture visualization, which represented the malicious code through code mapping, texture segmentation and texture feature extraction in a new image texture feature space. Pinhero et al. [15] visualized malware into grayscale, color and single-channel Markov images and fused Gabor filters to extract features of three images to achieve malware classification. Fu et al. [26] visualized malware as RGB images and introduced entropy information as part of feature representation, which effectively improved malware classification. Literature [27,28] visualized malware as grayscale images and extracted GLCM features to detect malicious code, and the experimental results showed that GLCM could extract texture and spatial information of images. Kumar et al. [17] exploited the advantages of transfer learning to quickly identify malware visualization features by pre-training the model, improving performance and learning efficiency. Notably, Conti et al. [29] visualized multiple features extracted from malware binaries, which opened up a new direction for applying visualization techniques in malware detection. Inspired by these research results, we propose a malware document visualization method that fuses the entropy information of the document, the texture and spatial information extracted by GLCM and the raw byte information of the document to solve the problems of insufficient feature characterization capability in malware document detection, cumbersome and complex feature engineering, time-consuming detection process that is difficult to be used for large-scale detection, as well as the problem of detecting documents in a single format only.

3 Proposed Method

Malicious document developers often modify existing code to create new variants of malicious documents. If we try to represent these documents with images, advanced malicious document visualization techniques can capture these subtle changes. The framework of this paper's scheme is shown in Fig. 1.

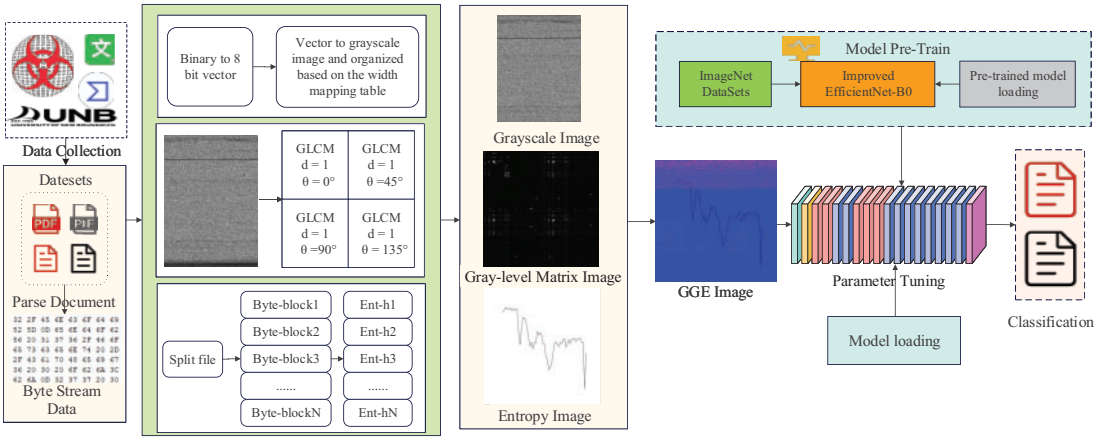

Figure 1: System architecture

3.1 Document Visualization Methods

3.1.1 Grayscale Image

The grayscale image is a technique to convert the original binary document to an 8-bit binary number and generate a grayscale image. As shown in Fig. 2, the binary document is first read according

to the 8-bit unsigned integer and each byte is considered as a pixel, in which each pixel takes a value ranging from 0 to 255 (black is represented by 0, and white is represented by 255). The pixel value is calculated as follows:

$$B_{bin} = b_7 \times 2^7 + b_6 \times 2^6 + b_5 \times 2^5 + b_4 \times 2^4 + b_3 \times 2^3 + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0 \quad (1)$$

$$H_{hex} = h_1 \times 16^1 + h_0 \times 16^0 \quad (2)$$

$(b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0)$, which is a binary sequence, (h_1, h_0) is a hexadecimal sequence.

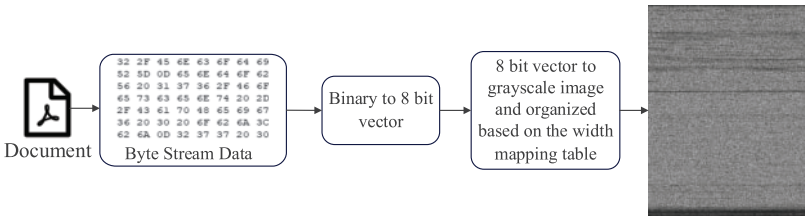

Figure 2: Flow of generating a grayscale image

As shown in Fig. 3, when malicious and benign documents are converted into grayscale images, they will show their specific texture characteristics. Malicious documents often contain specific code, encrypted data, hidden metadata, etc., presented in a specific texture when converted into an image. In contrast, benign documents usually contain regular content such as text, images, and tables, which have more straightforward and more regular texture distributions and, when converted to images, display relatively uniform and consistent texture characteristics.

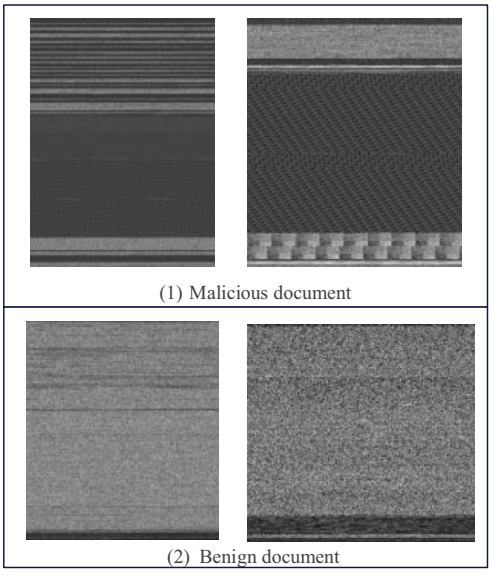

Figure 3: Example of a document grayscale image (1) Malicious document, (2) Benign document

In addition, it is essential to facilitate subsequent image analysis and deep learning algorithm model processing while considering the extensive range of document size span and ensuring the appropriate aspect ratio of the mapped grayscale image. In this paper, we develop a grayscale image

width mapping table based on the document size, as shown in Table 1, and using this mapping table and the byte size of the document, we can calculate the height of the image, which in turn generates a grayscale image. Since the documents vary in size, they are converted into grayscale images showing different sizes. In order to facilitate the subsequent input to the neural network for processing and analysis, in this paper, the grayscale images obtained from the mapping of malicious documents are normalized to ensure that all the images are uniformly sized at 256 × 256 pixels.

Table 1: Table of visualization width settings for byte-sequence data

File size	Image width
<10 KB	32
10∼30 KB	64
30∼60 KB	128
60∼100 KB	256
100∼200 KB	384
200∼500 KB	512
500∼1000 KB	768
>1000 KB	1024

3.1.2 Grayscale Matrix Image

The grayscale level co-occurrence matrix is a commonly used feature extraction method for image texture analysis, which describes texture information by capturing the joint distribution of gray levels of two pixels of an image. The core of this method lies in counting the frequency of gray-level co-occurrence between two pixels in a particular direction in an image to generate a co-occurrence matrix, which is calculated as follows:

$$P(i,j|d,\theta) = \#\{(x,y)\,|f(x,y) = i, f(x+dx, y+dy) = j; x, y = 0, 1, 2, \ldots, L-1\} \tag{3}$$

where # denotes the set; d is the relative distance expressed in terms of the number of pixels; θ denotes the four directions, 0°, 45°, 90°, and 135°; (x, y) are the coordinates of the pixels in the image, $i, j = 0, 1, 2, \ldots, L-1$; and L is the number of gray levels in the image.

In this paper, we adopt an innovative approach to grayscale level co-occurrence matrices, which is different from the previous approaches of calculating statistical features (e.g., contrast, energy, homogeneity, and correlation) directly from the GLCM, and we extract the four-direction grayscale level co-occurrence matrices through the GLCM to generate a new grayscale matrix image. Fig. 4 summarizes the grayscale matrix image generation process, Fig. 5 provides an example of a grayscale matrix image, and the Algorithm 1 describes the process of extracting GLCM image. The specific steps are as follows:

a) Grayscale image generation: Each document is converted into a grayscale image, which was generated in the previous subsection.
b) Grayscale level quantization: To reduce computational complexity and speed up the generation of GLCM, we use a grayscale level downgrade operation to quantize the original 256 gray levels into 128 levels.

c) Grayscale level co-occurrence matrix computation: Based on the 128-level gray-level image, we computed the gray-level covariance matrix with an offset of 1 in four directions (0°, 45°, 90°, and 135°). For each direction, a 128 × 128 sub-matrix was generated.
d) Grayscale matrix image generation: The sub-matrix for the 0° direction is placed in the upper left corner, the sub-matrix for the 45° direction is placed in the upper right corner, the sub-matrix for the 90° direction is placed in the lower left corner, and the sub-matrix for the 135° direction is placed in the lower right corner. In this way, we integrate the four sub-matrices with different orientations into a 256 × 256 matrix, which is finally visualized as a grayscale matrix image of 256 × 256 pixels.

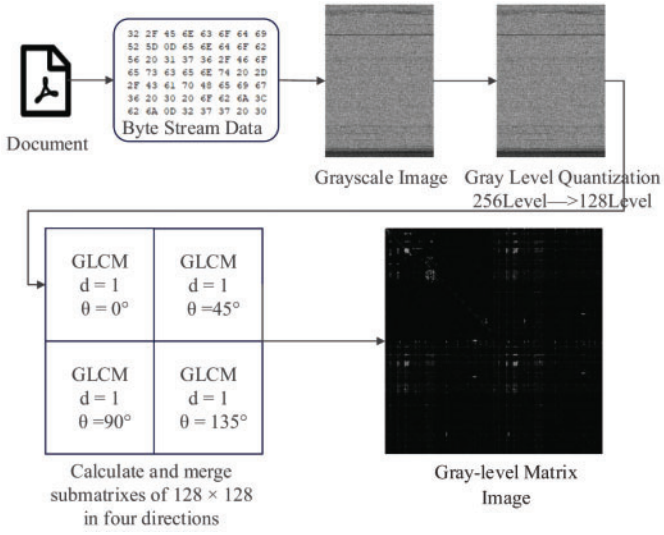

Figure 4: Grayscale level matrix image generation process

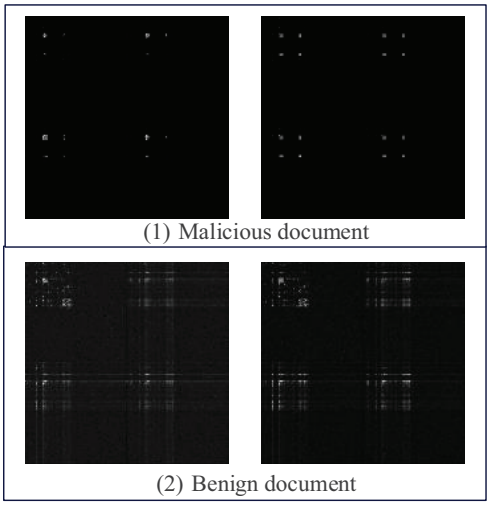

Figure 5: Example of a document grayscale level matrix (1) Malicious document, (2) Benign document

Algorithm 1: Grayscale level co-occurrence matrix images

Input: Document gray images L_i; angles; distance; maxGrayLevel;
Output: GLCM matrix combined GLCM, glcmImage;

```
For each l_i in L_i do;
  inputImage = loadImage(l_i);
  reducedImage = reduceLevels(inputImage, maxGrayLevel);
  for each angle in angles do;
    glcm = calculateGLCM(reducedImage, angle, distance);
    glcms.append(glcm);
  end for;
  combinedGLCM = combineGLCMs(glcm0, glcm45, glcm90, glcm135);
  glcmImage = glcmMatrixToImage(combinedGLCM);
End For
```

We convert the traditional GLCM and its stored texture and spatial information features into image form, enhancing the intuition of texture analysis and facilitating subsequent processing by deep learning algorithms. In this way, we can capture texture features in document images more effectively, enhancing the classification of malicious documents.

3.1.3 Entropy Image

In order to realize the conversion of the document to an entropy image, inspired by the work method of literature [29], this paper divides the document's content according to each block of 256 bytes. Then it calculates the Shannon entropy of each data block, respectively, plots the entropy value of these data blocks into an image, in which the X-axis represents the bytes of data blocks. The Y-axis represents the Shannon entropy of the corresponding data blocks. The generation process is shown in Fig. 6, and the entropy image map generation algorithm is shown in Algorithm 2. If the document length is not a multiple of 256 bytes, we fill in the last piece by filling the empty bytes to ensure that all the data blocks are the same size. Finally, the entropy value of each data block is connected to form the information entropy sequence according to the order of the chunks. The method of calculating information entropy is shown in Eq. (4).

$$H(B) = -\sum_{i=0}^{255} p(b_i) * \log_2 p(b_i) \tag{4}$$

where b_i is the document byte data, $p(b_i)$ represents the probability of the ith byte in the data block, $H(B)$ represents the information entropy of a given data block. The range of values is [0, 8]. Suppose the document is divided into N chunks; the entropy sequence is expressed as $H = h_1, h_2, \ldots, h_N$. Fig. 7 shows the images of malicious and benign documents after converting them into entropy sequences.

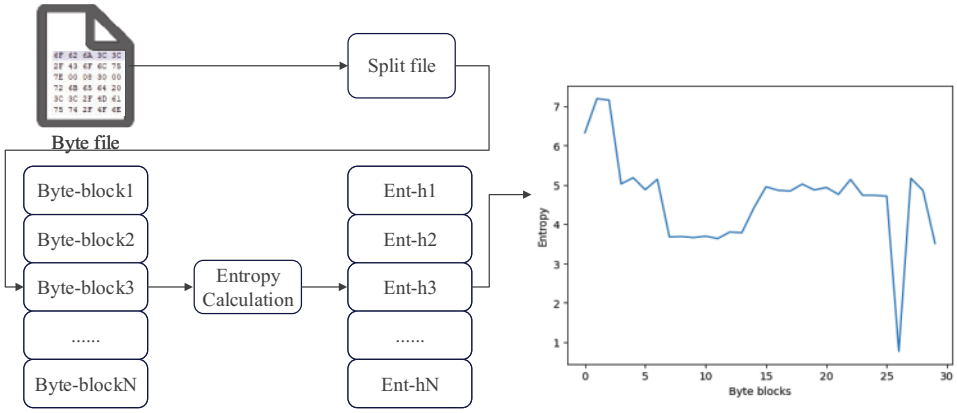

Figure 6: Entropy image generation process

Algorithm 2: Entropy map generator

Input: Malicious and benign document datasets D_i; *block_size*;
Output: Entropy graph *entropy_matrix*;

```
For each d_i in D_i do    // read documents and chunk them
  data=open(d_i, 'rb').read();
  for i in range(0, ceil(len(data)/block_size))
    block = data[i:i+block_size];
    if len(block)<block_size then
      block += ['0'] * (block_size - len(block))  // fill the last Block with zeros;
      end if
      blocks.append(block);
    end for
      for each block in blocks do
        entropy = calculateShannonEntropy(chunk);
       entropyValues.append(entropy);
      end for
      image = createImageFromEntropyValues(len(entropyValues), entropyValues);
      entropy_matrix = np.asarray(Image.open(image).convert('L'));
      return entropy_matrix;
End For
```

3.2 GGE Image Construction

We fused three different types of grayscale images into a three-channel image. Since the size of each type of image is 256 × 256 pixels, an empty array of shapes 3 × 256 × 256 is created as a basis for the fused image. In this array, we sequentially place a grayscale image, a grayscale matrix image, and an entropy image, each occupying a channel index. This way, we get a three-channel image called GGE (Grayscale Image, Grayscale Matrix Image, Entropy Image). The advantage of GGE image is the fusion of different image features into RGB color image and the ability to utilize existing migration learning models. An example of a GGE image is given in Fig. 8.

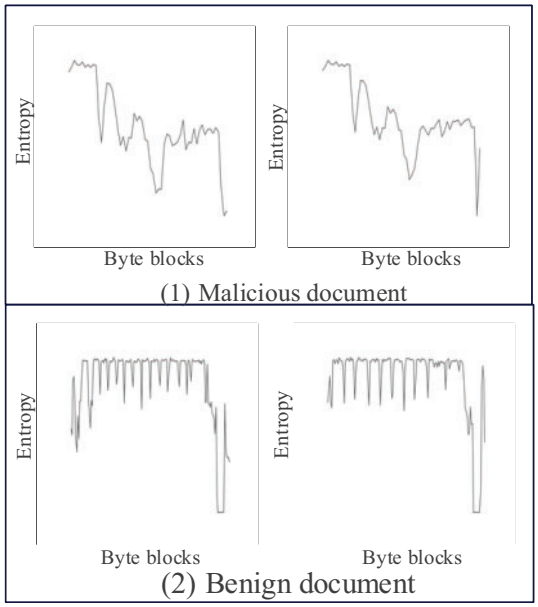

Figure 7: Example of a document entropy image (1) Malicious document, (2) Benign document

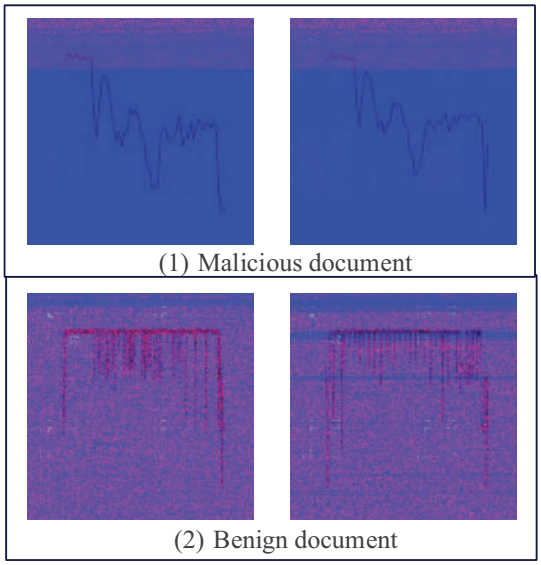

Figure 8: Example of a GGE image (1) Malicious document, (2) Benign document

3.3 Improved EfficientNet-B0 Model Design

EfficientNet-B0 is the base model in the EfficientNet family, which employs advanced architectural designs, including the MBConv (Mobile Inverted Bottleneck Convolution) and SE (Squeeze-Excitation) modules, which enable the model to achieve efficient feature extraction and representation capabilities while maintaining a small size. Specifically, EfficientNet-B0 uniformly adjusts the depth, width, and input image resolution of the network through a composite scaling method, which makes it

possible to greatly reduce model parameters and computation while maintaining accuracy. In addition, EfficientNet-B0 has been validated on several well-known datasets, including ImageNet, CIFAR-100, etc., demonstrating its excellent generalization ability and adaptability. In addition, it has also demonstrated excellent performance in several computer vision tasks, such as object detection and image segmentation, proving its strong generalization ability and usefulness. Therefore, in this paper, the EfficientNet-B0 pre-trained model is chosen as the feature extractor.

3.3.1 MBConv Module

This paper adopts the EfficientNet-B0 pre-trained model as a feature extractor, which is structured by the following parts: a Stem layer for initial feature extraction containing convolutional operations, normalization and activation functions; followed by 16 MBConv modules constituting the main body of the network; and after that the Conv2D, Pooling layer and fully connected layer. The core structure of the network is the MBConv module, which contains depth-separable convolution and Swish activation functions and integrates the SE attention mechanism. The structure of the MBConv module is shown in Fig. 9.

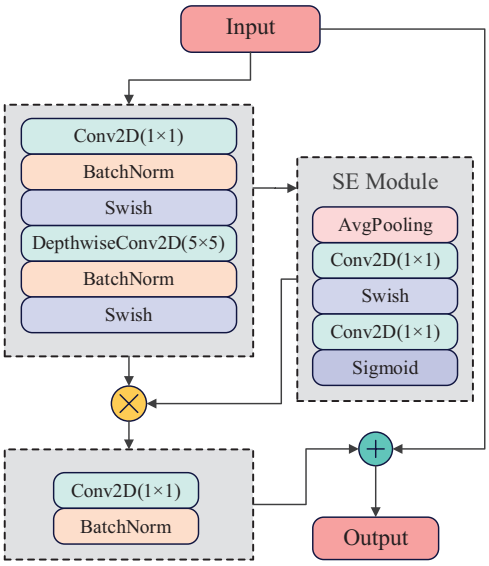

Figure 9: Structure of MBConv module

3.3.2 Fusion CBAM Module

CBAM (Convolutional Block Attention Module) consists of two key components: Channel Attention Module (CAM) and Spatial Attention Module (SAM). The CBAM module is lightweight and highly universal. It can be seamlessly integrated into various convolutional neural network architectures, which can comprehensively and meticulously pay attention to the input data's critical information, thus improving the model's accuracy and robustness. The structure of CBAM is shown in Fig. 10.

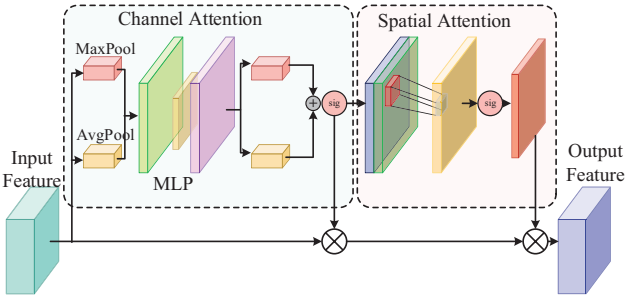

Figure 10: CBAM module structure

The input features are first processed through the global average pooling layer and maximum pooling layer to capture the features' spatial and local contextual information, respectively. Next, these features are processed through the shared MLP layer for element-by-element summing operation, and finally, the channel attention feature map is generated using Sigmoid. The channel attention module formula can be expressed as:

$$\begin{aligned} M_c(F) &= \sigma\left(MLP\left(AvgPool\left(F\right)\right) + MLP\left(MaxPool\left(F\right)\right)\right) \\ &= \sigma\left(W_1\left(W_0\left(F^c_{avg}\right)\right) + W_1\left(W_0\left(F^c_{max}\right)\right)\right) \end{aligned} \tag{5}$$

where W_0 and W_1 are the weights of the MLP, F^c_{avg} and F^c_{max} represent the average and maximum pooling features, respectively. σ represents the Sigmoid function.

The input feature maps are first processed by global average and maximum pooling, and the two pooling results are superimposed to reduce the dimensionality of the feature maps using 1×1 convolution to generate spatial attention feature maps using Sigmoid. The spatial attention module formula is expressed as:

$$\begin{aligned} M_s(F) &= \sigma\left(f^{(7\times7)}\left(\left[AvgPool(F); MaxPool(F)\right]\right)\right) \\ &= \sigma\left(f^{(7\times7)}\left(\left[F^s_{avg}; F^s_{\max}\right]\right)\right) \end{aligned} \tag{6}$$

where $f^{(7\times7)}$ represents the convolutional kernel size.

The channel and spatial attention feature maps are multiplied element-by-element with the original input feature maps to enhance key spatial regions in the feature maps and obtain the final output feature maps.

$$M_{out} = F \otimes \left(M_c\left(F\right) \oplus M_s\left(F\right)\right) \tag{7}$$

where $\oplus$ represents element-by-element addition and $\otimes$ represents element-by-element multiplication.

Although the channel attention mechanism SE is already included in EfficientNet-B0, the CAM module in CBAM optimizes and improves SE. In addition, the original network ignores spatial information, which CBAM aptly complements. Therefore, incorporating CBAM into the EfficientNet-B0 network can enhance the network's ability to extract features and thus improve recognition accuracy.

As shown in Fig. 11, the improved EfficientNet-B0 model consists of EfficientNet-B0 and CBAM, which we named the C-EfficientNet model. EfficientNet-B0 is responsible for the initial extraction of features, while the CBAM module is embedded after the 16 MBConv modules for further refinement of these features.

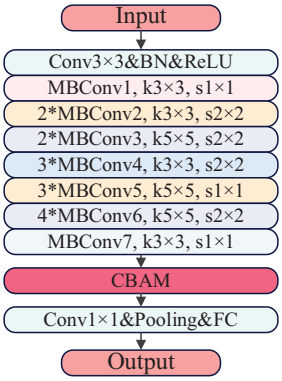

Figure 11: C-EfficientNet model

When the input feature map is fed into the CBAM module, the CAM module first processes it to generate the channel attention feature map M_c. This process enhances the feature map F by emphasizing the important channel features. Secondly, M_c is multiplied with the original feature map F to get the channel attention refined feature map F'. Subsequently, F' is fed into the SAM module to generate the spatial attention feature map M_s, which is responsible for extracting and emphasizing the spatial information in the feature map. Finally, M_s is multiplied by F' to obtain the final refined feature map, which incorporates both channel and spatial attention information and provides richer feature support for subsequent classification tasks.

Precisely, in our study, each sample is provided to the CBAM-EfficientNet-B0 model as a $3 \times 256 \times 256$ image. After in-depth processing by multiple convolutional and pooling layers in the model, we extract a 1280-dimensional feature vector. This vector is enriched with high-level semantic information of the samples, which is subsequently processed through a fully connected layer for classification and detection of malicious documents.

4 Experimentation and Evaluation

4.1 Experimental Environment and Dataset

The experiments are conducted using Ubuntu 18.04, an operating system with an Intel(R) Xeon(R) Gold 5218 CPU @ 2.30 GH processor, an NVIDIA GeForce RTX 3090 GPU, and 256 GB of RAM. The Pytorch deep learning framework and Python 3.7 are used.

In this paper, we used PDF and office document (including DOC, XLS, PPT, DOCX, XLSX, and PPSX formats) datasets, of which the PDF document data came from Contagio (collecting 3443 malicious documents and 4532 benign documents) and Evasive-PDFMal2022 (collecting 5557 malicious documents and 4468 benign documents) public datasets, collecting 9000 malicious and benign PDF documents. Malicious office documents were obtained from well-known malware vendors VirusShare and VirusTotal, while benign office documents were obtained from Baidu Wikipedia, with 1724 malicious and benign office documents collected. All documents were scanned and detected by VirusTotal before using in order to monitor their execution and ensured that malicious documents have malicious behaviors and benign documents had no malicious behaviors. Both datasets were divided into training and test set in the ratio of 8:2, the network parameters were optimized using the cross-entropy loss function and Adam algorithm, the learning rate was dynamically adjusted using

MultiStepLR, and the size of the input image was 256 × 256 × 3. And the configuration of the parameters during training is shown in Table 2.

Table 2: The parameters during training

Parameter	Value
Learn rate	0.001
Batch size	64
Epoch	50
Optimizer	Adam
lr_scheduler	MultiStepLR
Loss function	Cross-entropy loss

4.2 Evaluation Indicators

In order to measure the performance of the model in this paper on the malicious document detection task, this paper used a variety of metrics widely used in malicious detection methods for performance evaluation, including Accuracy, Precision, Recall, F1-Score, and False Positive Rate (FPR). Where Accuracy was the ratio of the number of documents correctly classified by the model (including those correctly identified as malicious and those correctly identified as benign) to the total number of documents tested. The precision rate was concerned with the proportion of genuinely malicious documents out of those the model determines to be malicious. Recall measured the proportion of malicious documents that the model successfully identifies. The F1-Score was the reconciled average of the precision and recall rates and was used to evaluate the model's performance comprehensively. The False Positive Rate measured the proportion of all benign documents the model incorrectly determines to be malicious. Where TP represents the number of documents classified as benign and actually benign, FN represents the number of documents classified as malicious but actually benign, FP represents the number of documents classified as benign but actually malicious, and TN represents the number of documents classified as malicious and actually benign.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1 - Score = \frac{2\,precision \times recall}{precision + recall} \tag{11}$$

$$FPR = \frac{FP}{FP + TN} \tag{12}$$

4.3 Comparison of Classification Experiments for Each Model

To evaluate the performance of the models in this paper for malicious document detection, we conducted a series of comparative experiments on PDF and office document datasets using the GGE method. We selected ResNet-50, VGG19_bn, InceptionV3, DenseNet-161, ConvNeXt-Tiny, and EfficientNet-B0 models as the control. The parameters of these models were the optimal solutions derived from several experiments, and the experimental results were shown in Table 3.

Table 3: Comparison of detection of different models

Datasets	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)
PDF	ResNet-50	96.92	95.62	98.33	96.96	4.50
	VGG19_bn	98.86	98.83	98.89	98.86	1.17
	InceptionV3	98.92	98.78	99.06	98.92	1.22
	DenseNet-161	99.14	98.95	99.33	99.14	1.06
	ConvNeXt-Tiny	98.42	97.65	99.22	98.43	2.39
	EfficientNet-B0	99.17	98.79	99.56	99.17	1.22
	Proposed method	**99.44**	**99.06**	**99.88**	**99.47**	**0.94**
Office	ResNet-50	89.28	88.39	90.44	89.40	11.88
	VGG19_bn	93.91	94.17	93.62	93.90	5.80
	InceptionV3	95.22	95.88	94.49	95.18	4.06
	DenseNet-161	95.80	96.47	95.07	95.77	3.48
	ConvNeXt_Tiny	92.75	94.29	91.01	92.63	5.51
	EfficientNet-B0	96.38	97.06	95.62	96.34	2.90
	Proposed method	**97.39**	**97.95**	**96.81**	**97.38**	**2.03**

Through the comparative analysis in Table 3, we could notice that for both PDF and office datasets, the model designed in this paper has been improved in several evaluation metrics (including accuracy, precision, recall, and F1-Score) and had the lowest FPR. Compared with ResNet-50, VGG19_bn, InceptionV3, DenseNet-161, and ConvNeXt-Tiny models, the accuracy of this paper's model was improved by 2.53%, 0.58%, 0.53%, 0.31%, and 1.03% on the PDF dataset; the F1-Score was respectively improved by 2.51%, 0.61%, 0.55%, 0.33% and 1.04%. On the office dataset, the accuracy was improved by 8.12%, 3.48%, 2.17%, 1.59%, and 4.64%, and the F1-Score was improved by 7.98%, 3.48%, 2.19%, 1.61%, and 4.75%, respectively. The EfficientNet-B0 model performed better in the experiments. However, the C-EfficientNet model performed better, and these data suggested that the EfficientNet-B0 model was more suitable for malicious document visualization classification work than other models.

The improved model C-EfficientNet added the CBAM module compared to the EfficientNet-B0 model. In order to verify the effectiveness of the improved model, the C-EfficientNet model was compared with the baseline model EfficientNe-B0 on the PDF dataset; its accuracy, precision, recall, and F1-Score were improved by 0.28%, 0.28%, 0.33%, and 0.30%, respectively, while the FPR was reduced by 2.08%; on the office dataset, its accuracy, precision, recall, and F1-Score were improved by 1.01%, 0.89%, 1.19%, and 1.04%, respectively. In comparison, the FPR was reduced by 0.87%. This demonstrated that introducing the CBAM attention mechanism is crucial for spatial and channel

attention of features, which helped to integrate feature information more efficiently, thus improving the classification performance.

4.4 Comparison of Complexity of Models

When evaluating classification algorithms, accuracy, recall, F1-Score, and precision were essential indicators for measuring the performance of the model; other than that, the time complexity and space complexity of the model were also non-negligible indicators, which could effectively evaluate whether the model can effectively realize the detection of malicious documents in real-time. Time complexity refered to the number of operations of the model, which was directly related to the time needed in the training and prediction phases of the model and was quantified by the number of floating-point operations, Flops, the higher the value of Flops, indicating that the model had a more significant amount of operations. Accordingly, the time needed to complete the same classification task is also longer. Space complexity refers to the number of parameters of the model, which reflected the model's occupancy on storage and memory resources. Table 4 compared the complexity of each model in detail.

Table 4: Comparison of model complexity

Model	FLOPs	Parameters
ResNet-50	5.38GFlops	24,558,146
VGG19_bn	25.69GFlops	143,678,248
InceptionV3	3.86GFlops	21,789,666
DenseNet-161	10.21GFlops	26,476,418
ConvNeXt-Tiny	5.818GFlops	27,800,258
EfficientNet-B0	**519.67MFlops**	**4,010,110**
C-EfficientNet	519.69MFlops	4,023,008

As can be seen from Tables 3 and 4, compared with the ResNet-50, VGG19_bn, InceptionV3, and ConvNeXt-Tiny models, our model showed significant superiority in all evaluation indicators. Compared with the DenseNet-161 model, although the improvement of the C-EfficientNet model was not particularly obvious in terms of accuracy and other indicators, regarding the number of parameters, the number of parameters of the improved model was only 1/6.6 of that of the DenseNet-161. Regarding computational complexity, the computational amount of the DenseNet-161 was 10.21GFlops. The computational complexity of DenseNet-161 was 10.21GFlops, while that of the C-EfficientNet model was 519.69MFlops, which was only 1/20 of that of DenseNet-161. Compared with the EfficientNet-B0 model, the model of this paper was significantly more accurate than the EfficientNet-B0 model in detecting PDF and office documents. In addition, although the network parameters and computation of this paper's model were slightly higher than that of the EfficientNet-B0 model, these increases are within acceptable limits. Through experimental evaluation, this model demonstrated robustness and effectiveness and possesses lower complexity and higher computational efficiency than other network models.

4.5 Comparative Experiments on the Effectiveness of Different Feature Classification

In order to deeply explore the influence of different feature strategies on the classification effect, we used the model in this paper to compare the three methods of extracting features by visualization techniques, as well as the direct conversion of documents to RGB color images with the color images

of GGE, Figs. 12 and 13 showed the experimental results of various types of feature methods on PDF and office datasets, respectively. As can be seen from the figures, on both datasets, the use of GGE images demonstrated significant advantages in several evaluation metrics compared to methods that only used a single grayscale image or directly convert to RGB color images.

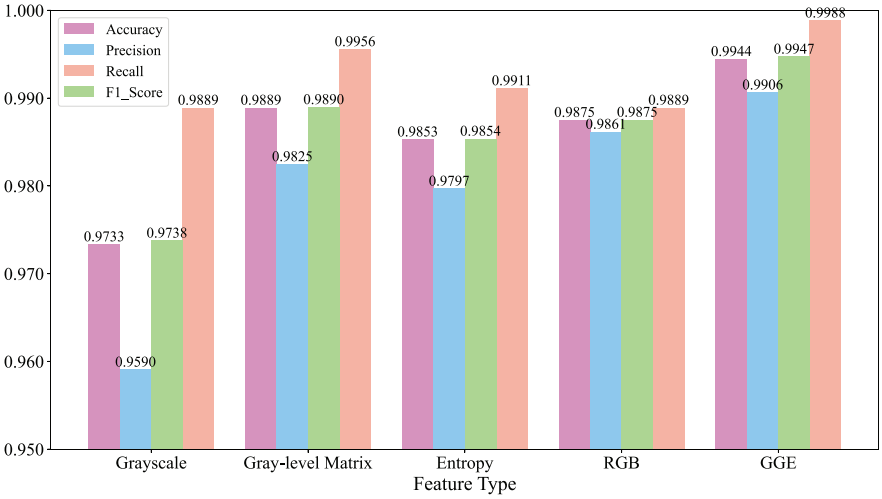

Figure 12: Experimental results of various types of feature methods on PDF dataset

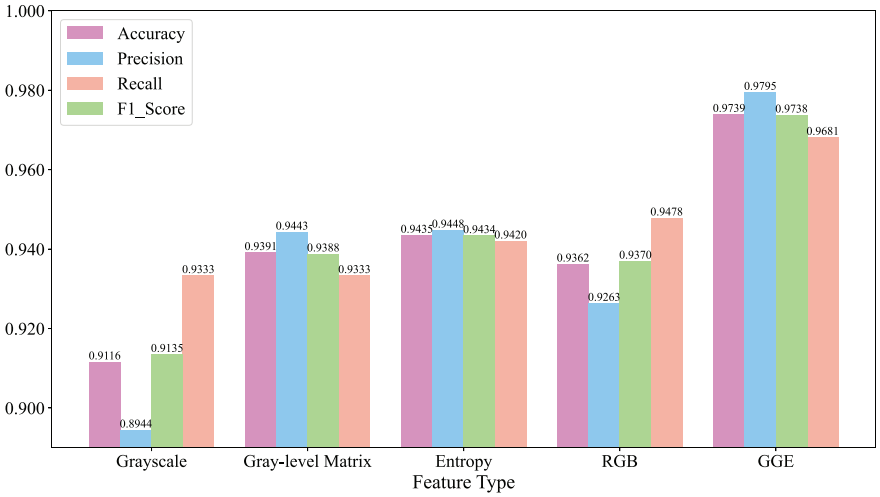

Figure 13: Experimental results of various types of characterization methods on the office dataset

It was worth noting that the use of the grayscale image method compared to other methods of classification effect was poor, which was mainly attributed to the existence of differences in the size of different documents, resulting in the conversion of grayscale images through scaling or cropping may be lost part of the information, which in turn affected the classification of the accuracy of the index and other indicators. In contrast, Entropy and GLCM grayscale image methods showed relatively good classification results, in which the accuracy of the GLCM method reached 98.90% on the PDF dataset, and the accuracy of the Entropy method reached 94.39% on the office dataset.

Furthermore, by fusing the grayscale image, Entropy and GLCM to construct a GGE color image, this innovative method achieved further enhancement in classification accuracy, reaching 99.44% on the PDF dataset and 97.39% on the office dataset, which proved the effectiveness of multi-feature fusion in improving the classification performance. Compared with the method of directly converting documents to grayscale images, the accuracy of the RGB color image-based method was improved by 1.42% and 2.46% on the PDF and office datasets, respectively. This was because RGB images contain three channels of information per pixel point and, therefore, contain more information, which could reflect the details and features of the document more comprehensively. The GGE method improved the accuracy by 0.69% and 3.77% on PDF and office datasets, respectively, compared to the RGB method. This was because the GGE image not only inherits the advantage of multi-channel information of RGB image but also incorporates the byte sequence information, entropy sequence information of the document, as well as the texture and spatial information stored in the GLCM, which formed a more comprehensive and deeper image characterization and provided a richer feature support for the document classification task.

4.6 Comparative Experiments with Other Existing Methods

In order to verify the effectiveness of the methods in this study, we compared them with the methods proposed in the literature [11,12,19,20,25], and Table 5 showed the results of the comparison experiments. Among them, Hidost [19] and 3SPDF [25] focued on the detection of PDF documents, while NFDD [11] and SFEM [20] were targeted at office documents. 3SPDF extracted features from the structure of PDF documents then chunked the documents, calculated the entropy sequences, and extracted the entropy sequences' statistical features using the wavelet energy spectrum. Hidost used the idea of abstraction to extracted the metadata features of PDF documents and described the structural features of the documents through the structural path method. NFDD used the Cuckoo sandbox to perform a 1-min dynamic analysis of the document, extracted the high-frequency word sequences in the report, and used the TextCNN model to train to obtain the classification results. SFEM took advantage of the hierarchical properties of office documents to convert the structural path of the document into a unique one. ESRMD not only extracted the statistical features of document information entropy sequences but also extracted the structural features of information entropy sequences by combining the Haar Wavelet Transform and Bag of Words model algorithm, and finally classifies them using machine learning algorithms.

Table 5: Comparison experiments with other methods

Method	Formats	Time (s)	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
3SPDF [25]	PDF	3.87	98.77	98.75	98.75	98.74
Hidost [19]	PDF	5.23	97.72	99.67	98.94	97.75
Our method	PDF	**3.07**	**99.06**	**99.88**	**99.47**	**99.44**
NFDD [11]	office	62.32	**98.16**	97.67	**97.91**	**97.92**
SFEM [20]	office	**2.16**	94.32	96.23	95.27	95.22
ESRMD [12]	office	10.58	96.10	**98.10**	97.08	97.11
Our method	office	3.73	97.95	96.81	97.38	97.39

As can be seen from Table 5, in PDF document detection, the GGE method proposed by us performs the best; its accuracy was 0.7% and 1.69% higher than that of the literature [19,25], respectively, and also had a significant advantage in the detection speed, the average detection of a PDF document only took 3.07 s.

The performance of GGE and ESRMD was quite similar regarding office document detection. However, GGE was more advantageous regarding detection speed, with an average time of 3.73 s to detect an office document, while ESRMD took 10.58 s. Although NFDD slightly outperformed our method in terms of accuracy, recall, precision, and F1-Score, our method was still within the acceptable range in terms of overall performance. It was worth noting that NFDD obtained the analysis report and extracted feature information by sandboxing the dynamic analysis for 1 min, so it took significantly longer to detect a document, about 16.7 times longer than GGE. In contrast, although SFEM was 1.57 s faster than the present method in terms of detection speed, it was 2.17 percentage points and 3.63 percentage points lower in terms of accuracy and precision, respectively, and SFEM needed to perform feature extraction manually, while the present method did not need this step.

Overall, GGE performed superiorly compared to other methods. It was suitable for detecting malicious documents in different formats, and it took less time in the detection process with a higher accuracy rate, which could be effectively applied to the task of detecting malicious documents in real-time.

5 Conclusion

In this paper, we introduce a novel malicious document visualization method, GGE image, which enhances the detection accuracy of malicious documents by combining grayscale, grayscale matrix, and entropy maps, and combines it with a deep learning model to improve the classification performance. Experimental results show that the CBAM-EfficientNet-B0 model incorporating GGE images achieves 99.44% and 97.39% accuracy on PDF and Office document datasets, respectively, demonstrating good generalization ability and robustness. Compared with the existing methods, the method in this paper is superior in detection performance, solves the problems of complex feature engineering, time-consuming detection process and difficulty used for large-scale detection, as well as the detection scope is limited to a single document format, which provides a new direction for the research of malicious document detection and is of great significance for improving the effect of malicious document classification in the future.

6 Future Work

In future work, our research direction will focus on two key areas: one is to explore the document information entropy sequence visualization feature extraction method. The second is to study in depth the anti-attack techniques in malicious document detection and ensure efficient and accurate detection performance in the complex and changeable network environment by enhancing the robustness and resilience of the model.

Acknowledgement: The authors would like to thank all the reviewers who participated in the review.

Funding Statement: This work has been supported by the Natural Science Foundation of Henan Province (Grant No. 242300420297) awarded to Yi Sun.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Youhe Wang; data collection: Yujie Li; analysis and interpretation of results: Youhe Wang, Chuanqi Zhou; draft manuscript preparation: Youhe Wang, Yi Sun. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Yi Sun, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] C. Lin, "New Banking Trojan "CHAVECLOAK"," Targets Brazil. 2024. Accessed: Oct. 8, 2024. [Online]. Available: https://www.fortinet.com/blog/threat-research/banking-trojan-chavecloak-targets-brazil

[2] R. Lakshmanan, "New phishing attack uses clever microsoft office trick to deploy NetSupport RAT," 2024. Accessed: Oct. 28, 2024. [Online]. Available: https://thehackernews.com/2024/03/new-phishing-attack-uses-clever.html

[3] Check Point Team, "PDF-based email threats on the rise: Fight back with AI-powered prevention," 2024. Accessed: Oct. 24, 2024. [Online]. Available: https://blog.checkpoint.com/harmony-email/pdf-based-email-threats-on-the-rise-fight-back-with-ai-powered-prevention

[4] A. Falah, L. Pan, S. Huda, S. R. Pokhrel, and A. Anwar, "Improving malicious PDF classifier with feature engineering: A data-driven approach," *Future Gener. Comput. Syst.*, vol. 115, no. 2, pp. 314–326, 2021. doi: 10.1016/j.future.2020.09.015.

[5] Y. N. Cui, Y. Sun, J. Luo, Y. H. Huang, Y. X. Zhou and X. L. Li, "MMPD: A novel malicious pdf file detector for mobile robots," *IEEE Sens. J.*, vol. 22, no. 18, pp. 17583–17592, 2020. doi: 10.1109/JSEN.2020.3029083.

[6] Y. -S. Jeong, J. Woo, and A. R. Kang, "Malware detection on byte streams of pdf files using convolutional neural networks," *Secur. Commun. Netw.*, vol. 2019, no. 1, 2019, Art. no. 8485365. doi: 10.1155/2019/8485365.

[7] H. Guo, S. Huang, C. Huang, Z. Pan, and M. Zhang, "File entropy signal analysis combined with wavelet decomposition for malware classification," *IEEE Access*, vol. 8, pp. 158961–158971, 2020. doi: 10.1109/ACCESS.2020.3020330.

[8] A. M. Zhou, L. Hu, L. P. Liu, P. Jia, and L. Liu, "Malicious office document detection technology based on entropy time series," (in Chinese), *J. Shandong Univ.(Nat Sci)*, vol. 54, no. 5, pp. 1–7, 2019.

[9] C. Carmony, X. Hu, H. Yin, A. V. Bhaskar, and M. Zhang, "Extract me if you can: Abusing PDF parsers in malware detectors," in *Proc. NDSS*, San Diego, CA, USA, 2016, pp. 1–15.

[10] M. Xu and T. Kim, "PlatPal: Detecting malicious documents with platform diversity," in *Proc. USENIX Security*, Vancouver, BC, Canada, 2017, pp. 271–287.

[11] J. G. Jiang *et al.*, "NFDD: A dynamic malicious document detection method without manual feature dictionary," in *Proc. WASA*, Nanjing, China, 2021, pp. 147–159.

[12] L. Liu, X. He, L. Liu, L. Qing, Y. Fang and J. Liu, "Capturing the symptoms of malicious code in electronic documents by file's entropy signal combined with machine learning," *Appl. Soft Comput.*, vol. 82, 2019, Art. no. 105598. doi: 10.1016/j.asoc.2019.105598.

[13] H. Guo, S. G. Huang, C. Huang, F. Shi, M. Zhang and Z. Pan, "Binary file's visualization and entropy features analysis combined with multiple deep learning networks for malware classification," *Secur. Commun. Netw.*, vol. 2020, no. 1, 2020, Art. no. 8881760. doi: 10.1155/2020/8881760.

[14] H. Deng, C. Guo, G. Shen, Y. Cui, and Y. Ping, "MCTVD: A malware classification method based on three-channel visualization and deep learning," *Comput. & Secur.*, vol. 126, no. 7, 2023, Art. no. 103084. doi: 10.1016/j.cose.2022.103084.

[15] A. Pinhero *et al.*, "Malware detection employed by visualization and deep neural network," *Comput. & Secur.*, vol. 105, no. 12, 2021, Art. no. 102247. doi: 10.1016/j.cose.2021.102247.

[16] Y. Zhao, C. Xu, B. Bo, and Y. Feng, "MalDeep: A deep learning classification framework against malware variants based on texture visualization," *Secur. Commun. Netw.*, vol. 2019, no. 1, 2019, Art. no. 4895984. doi: 10.1155/2019/4895984.

[17] S. Kumar, B. Janet, and S. Neelakantan, "IMCNN: Intelligent malware classification using deep convolution neural networks as Transfer learning and ensemble learning in honeypot enabled organizational network," *Comput. Commun.*, vol. 216, no. 1, pp. 16–33, 2024. doi: 10.1016/j.comcom.2023.12.036.

[18] F. Rustam, I. Ashraf, A. D. Jurcut, A. K. Bashir, and Y. B. Zikria, "Malware detection using image representation of malware data and transfer learning," *J. Parallel Distr. Comput.*, vol. 172, no. 3, pp. 32–50, 2023. doi: 10.1016/j.jpdc.2022.10.001.

[19] N. Šrndić and P. Laskov, "Hidost: A static machine-learning-based detector of malicious files," *EURASIP J. Inf. Secur.*, vol. 2016, no. 1, pp. 1–20, 2016. doi: 10.1186/s13635-016-0045-0.

[20] A. Cohen, N. Nissim, L. Rokach, and Y. Elovici, "SFEM: Structural feature extraction methodology for the detection of malicious office documents using machine learning methods," *Expert. Syst. Appl.*, vol. 63, no. 2–3, pp. 324–343, 2016. doi: 10.1016/j.eswa.2016.07.010.

[21] Y. Li, X. Wang, Z. Shi, R. Zhang, J. Xue and Z. Wang, "Boosting training for PDF malware classifier via active learning," *Int. J. Intell. Syst.*, vol. 37, no. 4, pp. 2803–2821, 2022. doi: 10.1002/int.22451.

[22] Q. Abu Al-Haija, A. Odeh, and H. Qattous, "PDF malware detection based on optimizable decision trees," *Electronics*, vol. 11, no. 19, 2022, Art. no. 3142. doi: 10.3390/electronics11193142.

[23] S. Y. Yerima and A. Bashar, "Explainable ensemble learning based detection of evasive malicious PDF documents," *Electronics*, vol. 12, no. 14, 2023, Art. no. 3148. doi: 10.3390/electronics12143148.

[24] W. P. Wen, Y. J. Wang, and Z. Meng, "PDF file vulnerability detection," (in Chinese), *J. Tsinghua Univ. (Sci Technolo)*, vol. 57, no. 1, pp. 33–38+43, 2017.

[25] J. G. Jiang *et al.*, "Detecting malicious pdf documents using semi-supervised machine learning," in *Adv. Digital Forensics XVII*, 2021, pp. 135–155. doi: 10.1007/978-3-030-88381-2_7.

[26] J. W. Fu, J. F. Xue, Y. Wang, Z. Y. Liu, and C. Shan, "Malware visualization for fine-grained classification," *IEEE Access*, vol. 6, pp. 14510–14523, 2018. doi: 10.1109/ACCESS.2018.2805301.

[27] I. T. Ahmed, B. T. Hammad, and N. Jamil, "A comparative performance analysis of malware detection algorithms based on various texture features and classifiers," *IEEE Access*, vol. 12, no. 19, pp. 11500–11519, 2024. doi: 10.1109/ACCESS.2024.3354959.

[28] Y. J. Liu, H. Fan, J. G. Zhao, J. F. Zhang, and X. X. Yin, "Efficient and generalized image-based CNN algorithm for multi-class malware detection," *IEEE Access*, vol. 12, no. 5, pp. 104317–104332, 2024. doi: 10.1109/ACCESS.2024.3435362.

[29] M. Conti, S. Khandhar, and P. Vinod, "A few-shot malware classification approach for unknown family recognition using malware feature visualization," *Comput. & Secur.*, vol. 122, no. 2, 2022, Art. no. 102887. doi: 10.1016/j.cose.2022.102887.